NLP and Sentiment Driven Automated Trading
Atish Davda, Parshant Mittal
Faculty Advisor: Michael Kearns

Abstract
• Perform syntax based sentence-level analysis on headlines of stock news, to determine trading feasibility.
• Infer word-sentiments from stock prices to trade stocks.

Final System Design
• Implemented in three phases, incrementally adding functionality; the final system diagram is shown below.

- The Parsed Data Source, written in Java for consistency with the remainder of the system, actually relies on Perl scripts which run periodically during the day and scrape Yahoo! Finance and Google Finance websites and gather news and stock price info. Parsed Data Source design:

- The Parsed Data Source, written in Java for consistency with the remainder of the system, actually relies on Perl scripts which run periodically during the day and scrape Yahoo! Finance and Google Finance websites and gather news and stock price info. Parsed Data Source design:

  Static HTML files are remnants of Phase 2 of the project, when we used three months of static news and price data.

  For greatest flexibility, allowing us to grab headlines for any given stock for any given date, we parsed news websites.

  Similarly, retro-parsed price info (open, close, volume) for any given date, storing both into the MySQL database.

  The natural next step in making the system live is integrating RSS feeds into the data source.

Sentiment Generation
• Data was gathered from July through November 2007. As you know these were turbulent times for all equities markets, as evident from the Market Return chart.
• Daily news and prices from July and August were used as the “learning bed” to assign sentiments (between -10 to 10) to corpus.

Frequency of corpus sentiments. Note: near 50/50 split between +/–
• Corpus
• Hand-selected 200 high-frequency words were “stemmed” and expanded, using WordNET, to over 2100 altogether.
  Stemming: using the Stanford Lex Parser’s stemming algorithm, reduce (falling, falls, fell, fail) to (fall).

NLP – Dependency Trees
• Extracted relationships between words, implied by syntax of headline using the Stanford Lex Parser. Once we classified each word by dependency tags (below) we built dependency trees.

  Tree for: “Cisco beats earnings forecasts.”

  Dependency tree structure: vertices are words, edges are relationships.

  Analysis begins at leaf nodes and then percolating up until the root. Edges have to be resolved via rules we defined. Example: conjunctions

  And

  “Google grows and soars.” Note: the number after the word is its sentiment.

  But

  “Google grows but fails.” Note: the number after the word is its sentiment.

  Difficult to apply on headlines – NOT sentences.

  Rules for many dependencies were manually developed by sampling headlines for ideas expressed within.

Trading Strategies & Results
• Various hypotheses were tested, including impact of no. of headlines, volume of trade, gain/loss on previous day.
• Predicted sentiment for each stock for each day were computed. BUY/SELL/HOLD correctness was measured by taking close-to-close return, i.e. taking a position.

  Data from November.

  As hypotheses tested, news, were times equities previous years.

  Correction: gained on previous year.

  Correctness by index.

  Returns by index during Oct, Nov 2007, and Strategy returns aggregated over indices:

  Aggregated Strategy Return

  Trading Strategies Measured

  Baseline: market returns by index during Oct, Nov 2007, and Strategy returns aggregated over indices:

  Bag-of-Words (PLTS20)

  Summary: two-month and annualized returns (averaged over the three indexes) for the different strategies.

Conclusions
• Even ignoring context, Bag-of-Words sentiment analysis yields promising results, +4.0% (12.7% over market).
• Sentiment-free momentum strategy yields -3.5% (i.e. momentum effect not too significant).
• Dependency analysis yields +3.5%, (12.2% over market), but no improvement over Bag-of-Words.
• NLP NOT INCREMENTALLY EFFECTIVE OVER SENTIMENTS because headlines do not follow standard grammatical rules.